FINAL REPORT

(Covering the period from
1 January 1974 to 31 December 1974)

NASA RESEARCH GRANT NGR23-005-622

THEORY OF RELIABLE SYSTEMS

Principal Investigator
John F. Meyer

April 1975

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Objectives

The purpose of this research project was to refine the current notion of system reliability by identifying and investigating attributes of a system which are important to reliability considerations, and to develop techniques which facilitate analysis of system reliability. Attributes selected for investigation included:

(a) Fault tolerance - the ability to maintain error-free input-output behavior in the presence of (temporary and/or permanent) faults in the system.

(b) Diagnosability - the ability to detect and locate faults in the system

(c) Reconfigurability - the ability to reconfigure the system after the occurrence of a fault so as to realize the original behavior or some other (possibly less complex) behavior

with the following objectives:

I. To determine, relative to the above attributes, properties of system structure that are conducive to a particular attribute. Structures so considered will range from state-transition functions at one extreme to hardware and software realizations at the other extreme.

II. To determine methods for obtaining reliable realizations of a given system behavior. In particular, one would like to obtain realizations which are fault tolerant (relative to the specified behavior) and yet diagnosable (relative to some extended behavior).

III. To determine how properties of system behavior relate to the complexity of fault tolerant (diagnosable, reconfigurable) realizations. Once such relationships are discovered, the inherent fault tolerance (diagnosability, reconfigurability) of a given behavior could be measured by the minimum complexity of realizations possessing that reliability attribute.

IV. To determine methods for evaluating the reliability of a proposed or existing system as measured in terms of fault tolerance, diagnosability, reconfigurability, or combinations of these attributes. This includes the investigation of appropriate reliability measures, modeling techniques, and computational methods for determining, or at least estimating, system reliability.

After initiation of the grant, the above proposed objectives were augmented to obtain a more definitive statement of what research should be accomplished to meet the needs of NASA and, in particular, the Langley Research Center. The following statements of these augmented objectives were due primarily to the constructive suggestions of NASA-Langley, with some subsequent modifi-

cation in wording to conform more closely with our interpretation:

I. To develop formal concepts and establish mathematical results which can be used to precisely define measures of system utility, e.g.:

1. Measures of fault tolerance;
2. Measures of recoverability based on measures of detectability, locatability and reconfigurability;
3. Measures of system availability with respect to different levels of system performance;
4. Measures of total system "worth" based on measures of performance worth and measures of performance availability.

II. To develop analytic and simulation methods for evaluating system utility measures.

III. To determine architectural characteristics of fault-tolerant systems that are amenable to fault detection and fault location.

IV. To investigate methods of on-line diagnosis that are applicable to specific subsystems of a fault-tolerant computing system, e.g.:

--given an arithmetic unit subject to a specified class of faults, design a detector that, with a

specified allowable time delay, will detect any error produced by a fault.

V. To investigate methods of augmenting the structure of specific hardware or software subsystems in order to facilitate detector design and improve on-line diagnosability.

1.2 Personnel

To meet the objectives stated in Section 1.1, it was estimated that the following technical effort would be required:

Principal Investigator
25 per cent time, academic year
100 per cent time, two months, summer

Research Assistants
1 at 50 per cent time, academic year
2 at 25 per cent time, academic
3 at 100 per cent time, summer

Programmer
25 per cent time, fiscal year

During the period 1 January - 31 December, 1974 (referred to as the "reporting period") research personnel and their level of effort have been:

Principal Investigator
John F. Meyer
25 per cent time, January - May
100 per cent time, June
25 per cent time, September - December

Research Assistants

David E. Frisque
20 per cent time, January - April
100 per cent time, May - July
25 per cent time, September - December

Carolyn P. Steinhaus
13 per cent time, February - April
100 per cent time, May - July
50 per cent time, September - December

Robert J. Sundstrom
54 per cent time, January - April
100 per cent time, May - July

1.3 Documentation

The following documents were produced during the reporting period.

Status Reports:

Semi-annual status report, NASA Research Grant NGR 23-005-622, July 1974

Technical Reports:

R. J. Sundstrom, "On-line diagnosis of sequential systems - II." Systems Engineering Laboratory Technical Report No. 81, The University of Michigan, Ann Arbor, July 1974.

R. J. Sundstrom, "On-line diagnosis of sequential systems - III." Systems Engineering Laboratory Technical Report No. 84, The University of Michigan, Ann Arbor, January 1975. (Part of final technical report)

Papers:

J. F. Meyer and R. J. Sundstrom, "On-line diagnosis of unrestricted faults," IEEE Transactions on Computers, Vol. C-24, No. 5, May 1975. (To appear)

J. F. Meyer, "Computation-based reliability analysis," Digest of the 1975 International Symposium on Fault Tolerant Computing, IEEE Computer Society, June 1975. (To appear)

Final Report:

J. F. Meyer, "Theory of reliable systems." Final technical report for NASA Grant NGR 23-005-622, Systems Engineering Laboratory Technical Report No. 83, The University of Michigan, Ann Arbor, January 1975.

2. SUMMARY OF TECHNICAL ACTIVITY

In keeping with the general objectives of the research project (see 1.1), several specific investigations were proposed for study during the reporting period. (See Proposal No. ORA 73-1167-KB1, "Theory of Reliable Systems," March, 1973). Early in the reporting period, it was decided that work during the year should focus on two of these investigations, namely:

(1) Reliability Analysis - Determine appropriate measures of system reliability that can be evaluated relative to some specified level of structural description, and develop models for reliability analysis, with respect to the above measures.

(2) On-Line Fault Diagnosis - Determine structural and behavioral properties of systems that are conducive to their "on-line" diagnosis; and determine methods for

altering the design of a system to improve its on-line diagnosability. As contrasted with "off-line" diagnosis, an on-line diagnostic procedure must contend with (i) system input over which it has no control and (ii) faults that occur as the system is being diagnosed.

With respect to each of these investigations, the technical report that follows describes:

(1) Background that motivated the activity;

(2) A summary of the results obtained;

(3) Topics for further investigation.

Detailed descriptions of research performed under the grant are contained in various technical reports and papers prepared during the reporting period (see 1.3). Collectively, the latter documents comprise a detailed final report of our research activity.

2.1 Reliability Analysis

2.1.1 Background

A review of the current state of the art of reliability analysis reveals a situation common to relatively new fields; namely, the tendency to hold on to concepts and methodologies that were introduced when the field first began to develop. Early objects of reliability analysis were pieces of electronic communication and control equipment whose functional requirements were relatively

easy to specify. Accordingly, what constituted "success" or "failure" of such systems was also easy to specify and usually directly related to the operation of physical components. Consequently, reliability measures such as "probability of success," "mean time to failure," "availability," etc., were unambiguous. Calculation of such measures followed naturally from information about the reliability of physical components.

During the past thirty years, however, the structural and functional complexity of man-made systems has increased tremendously, particularly in the computer field. Consequently, the analysis of reliability has become more complex. Considerable research effort has been devoted to developing formulae and computer programs to facilitate the calculation of traditional reliability measures for modern digital computers.

In general, this research has been based on formal models of the structure of the system being analyzed, and have implicitly carried along underlying notions of success and failure which are directly tied to the operation of structural components. (See [1]-[4], for example.) However, structure and function are no longer so closely related to each other in a modern digital computer. What we mean by "successful operation" of a computing system is a notion that is allied much more closely with function than with structure. In fact, "successful operation" is determined by (and dependent on) the functional require-

ments of each particular application which are likely to make distinct types of structural demands upon a machine.

For example, consider the differing requirements of the following uses of a large computer system: (1) a research and educational tool at a large university and (2) an automatic control and computation device aboard an aircraft or spacecraft (see [5], for example). In the first case, the demands upon the system are relatively constant over time and potentially utilize the entire system at any point in time. Furthermore, there is a very limited sense in which one could discuss "graceful degradation" of such a system. In the latter environment, however, requirements do vary with time, (for example, programs and data necessary for take-off may be unnecessary later), and some functions may be considerably less critical than others.

The example just cited is indicative of the need to more fully account for the behavior of a computer when analyzing its reliability. To accomplish this, reliability measures must refer to concepts of system success which involve more than just the status of various components or subsystems. The questions "Just what should be involved?" and "How is the involvement formalized?" were the primary questions that motivated our present activity in this area.

2.1.2 Present Activity

The purpose of the investigation summarized below was to give a precise meaning to the notion of a computation-based reliability analysis and indicate what kind of things must be considered if reliability measures are to more accurately reflect the computational needs of the user. Toward this end, we attempted to accomplish the following tasks during the reporting period:

(1) Develop a model of a "computer with faults" that can represent the effects of both permanent and transient physical failures, and permits the formal specification of computation-based success criteria.

(2) Using the model developed in (1), formally represent computation-based success criteria as "tolerance relations" on computations, and establish a precise notion of "success" (relative to a given tolerance relation).

(3) Using the concept of success developed in (2), formulate reliability measures that reflect the ability to rely on a system (when modeled as a computer with faults) in some specified use environment.

(4) Show how measures developed in (3) can be evaluated, and compare the results with those obtained using more conventional structure-based measures.

Research directed toward the accomplishment of each of these tasks has been conducted throughout the reporting period. Results obtained during the first half of the period were reported on in detail in the Semiannual Status Report [6]. During the second half of the period, concepts developed in connections with tasks (1) and (2) were refined where necessary, and work continued on tasks (3) and (4). Results of the total effort are described in the Technical Report "Computation-based Reliability Analysis" [7]. A paper with the same title, summarizing the Technical Report, has been accepted for presentation at the 1975 Symposium on Fault Tolerant Computing in Paris, France, June 1975 [8].

Briefly reviewing the results of this effort (as they relate to each of the four tasks described above):

(1) By modeling a (digital) computer as a discrete-time and, in general, time-varying system, it was shown that both permanent and transient (physical) failures can be represented by specifications (called "faults") for altering the prefailure transition function in an appropriate way. Given a computer C and a fault f, the computer with the altered transition function is denoted C^f (and called the "result of f"). A "computer with faults" was then defined as a triple (C,F,φ) where $C \in \mathscr{C}$ ($\mathscr{C}$ is the class of "computers"), F is a set of faults, and $\varphi: F \to \mathscr{C}$ where $\varphi(f) = C^f$. Associated

with each C $\varepsilon \mathscr{C}$ is a set of "state-behaviors" $\{\alpha_q | q$ is a state of C$\}$ and, in turn, a set of "computations" where each computation is a quadruple $(q,i,x,\alpha_q(x))$ with "initial state" q, "initial time" i, "input sequence" x and "state trajectory" $\alpha_q(x)$. Such computations are the basis for subsequently defined concepts of "success." (See [7], Section 3 for a detailed development of these results.)

(2) Computation-based success criteria were formally represented as reflexive relations (called "tolerance relations") on the set of all computations. Given a tolerance relation σ, the formalism developed in task (1) permitted the following precise definition of computational success: If u is a computation of computer C^f and u' is the corresponding computation of the fault-free computer C (i.e., u and u' have the same initial state, initial time, and input sequence) then u is a "σ-success" if u stands in the relation σ to u' . (See [7], Section 4.)

(3) Although we had hoped to formulate several reliability measures (or "utility" measures as they are referred to in the Langley-augmeted objectives), most of the effort here was devoted to the most basic reliability measure, "probability of success." Time constraints precluded a similar treatment of other measures such as "mean-time-to-failure," "availability," recoverability,"

etc., but we feel that these other measures can be dealt with in a similar fashion. Since "success," as defined in task (3), is the success of a computation, the probability of (system) success (in the use environment) requires a probability space for the computational environment as well as for the computer. Each of these spaces was formulated separately and then combined under the assumption that computational requirements are independent of faults. Letting P denote the measure for the combined space, the probability of σ-success of a computer C (called the "reliability of C") was formulated as P(H) where H is the set of all "environment-fault" descriptions (e,f) such that the computation $(e,\alpha_q^f(x))$ of C^f is a σ-success. (See [7], Section 5.)

(4) It was demonstrated how measures of the type developed in task (3) could be evaluated, using a read-only memory (ROM) as an example. For a specific set of assumptions regarding the probabilistic nature of memory faults and the computational environment of the ROM, the computation-based analysis yielded a reliability of .9904 while a conventional structure-based analysis yielded a reliability of .9680. (See [7], Section 6.)

2.1.3 Topics for Further Investigation

The results summarized above indicate that reliability analysis can indeed be formalized so as to reflect the ability of the user to rely on the computations a computer

performs (as opposed to the computer itself). It is also clear that the research performed to date is only a beginning in this direction, and there are a number of topics that deserve immediate investigation.

The first of these is the further exploration of examples that illustrate how the probability of (computation-based) success can be evaluated (see task (4), Section 2.1.2). We had hoped to do more of this during the present reporting period, but time did not permit.

Secondly, other reliability measures such as "mean-time-to-failure," "fault-tolerance," "availability" and "recoverability" should be given a computation-based formulation. In particular, recoverability (i.e., "coverage" [3]) should be focused on since it is inherently a computation-based measure. As remarked on in our detailed report [7], many structure-based models employ coverage as a parameter but cannot be used to determine the values of this parameter.

Third, there is need to consider specific instances of the general model that are closer to particular applications problems. The purpose of the general model has been to formalize reliability measures along with the information required to evaluate the measures. However, to obtain a useful tool for assessing the reliability of a special class of systems (e.g., aircraft computers), the model must be specialized to permit practical methods of evaluation using practically available data.

Fourth, the evaluation methods just referred to must be investigated. This includes simulation methods as well as analytic methods. When simulation methods are employed, the simulation should account not only for the probabilistic nature of faults but also for the probabilistic nature of the computational environment. The end product of this last investigation should be a set of algorithms which, given reasonable constraints on computer time, computer memory and cost, can evaluate a set of computation-based reliability measures.

The four topics outlined above are in close keeping with Langley-augmented objectives I and II (see Section 1.1) and, we believe, comprise a logical next-step in the development of meaningful and useful reliability assessment methods for aircraft computers.

2.2 Diagnosis

2.2.1 Background

In an increasingly large number of applications (e.g., communications switching, aircraft and spacecraft flight control, hospital patient monitoring) there is an obvious need for computers which are capable of operating for extended periods of time with extremely high reliabilities. Existing techniques for improving system reliability are traditionally divided into "passive" techniques (e.g., quadding,TMR) and "active" techniques

(e.g., stand-by sparing). The passive, or "fault-masking" techniques are frequently less complicated to implement, but a number of studies ([9] - [11]) have shown that it will be difficult if not impossible to achieve desired reliabilities--on a cost effective basis--through the exclusive use of passive techniques. Thus there seems to be widespread agreement that the future will bring ever wider use of active techniques.

Implied by such active techniques is the ability to detect a module which has failed and replace it with a standby spare before system failure occurs. In fact, many analyses of such systems (e.g., Mathur [1]) implicitly assume that failed modules are detected instantaneously. Such fault diagnosis techniques date back to the relay computers developed by Bell Laboratories in the early 1940's, where biquinary codes were used to dynamically check the operation of the computer. A general survey of the use of codes was made in 1959 by Peterson and Rabin [12], where they showed that combinational circuits can vary greatly in their inherent diagnosability.

Since then, the techniques that have been developed can be classified into two broad classes--off-line diagnosis and on-line diagnosis. Off-line diagnosis embraces those techniques where the system input is controlled. For example, running special diagnostic programs on a computer is a form of off-line diagnosis. On-line diagnosis, in

contrast, refers to diagnostic techniques which do not interfere with normal operation of the computer.

Obviously, on-line techniques are in general more desirable in real-time applications, where interruption of processing to run diagnostic programs is not possible. The use of coding schemes, parity bits, and the like are familiar on-line techniques. In addition, a number of special on-line diagnosis methods have been considered which apply to specific hardware subsystems, such as adders or counters (see [13], for example).

A theoretical study of on-line fault diagnosis was initiated under NASA Grant NGR 23-005-463. Our discussions with NASA-Langley convinced us of the need for improved on-line techniques which are suitable for incorporation into the architectures envisaged for computers designed for high-reliability applications.

The initial problem was to formulate an appropriate class of system models (i.e., a class of "systems with faults") that would serve as the basis for the study. It was decided that conventional models of time-invariant systems (e.g., sequential machines) are inadequate because they cannot represent the dynamics of a system which is experiencing faults. The representation finally selected was a class of resettable discrete time systems, which were adequate to represent both the structure and behavior of faulty and fault-free systems. In this formalism a fault f is represented by a triple $f = (S', \tau', \theta)$, with

the interpretation that a system S which experiences fault f is, as a result, transformed into system S' at time τ, with transient state behavior Θ. The result of f is the system S^f, which behaves like S up to time τ, and S' thereafter.

These systems were made resettable by including in the system description a reset function $\rho: R \times T \rightarrow Q$, where R is a finite nonempty set, T is the time base, and Q is the state set. This reset function has the interpretation that if reset r is applied to the system at time t-1, then the system will enter $\rho(r,t)$ at time t. Distinguishing the reset function is simply a matter of convenience. The same effect could have been achieved by incorporating an equivalent function into the general state transition function.

Once this model was selected, it was possible to formulate definitions for intuitive notions like fault tolerance, error, and diagnosability, in the overall context of on-line diagnosis. To summarize briefly, if S is a system and f is a fault of S, we say that f is tolerated if S^f mimics the behavior of some specified system $\tilde{S}$. Otherwise, f causes errors (i.e., erroneous behavior). Our notion of on-line diagnosis involves an external detector (assumed fault-free) which monitors system S. More specifically, if F is the set of all faults to which S is susceptible, then S is (D,k) diagnosable if, for all $f \in F$

(i) D responds negatively if S is fault-free,

(ii) D responds positively within at most k time steps following the occurrence of f.

The work outlined above is described in a rigorous fashion in [14].

2.2.2 Recent Activity

Activity during the reporting period was focused on investigating the diagnosis of two sets of faults: the set of "unrestricted faults" and the set of "unrestricted component faults."

The set of unrestricted faults of a system is simply the set of all possible faults of that system, that is $\{f|f \text{ is a fault of } S\}$. It is easily seen that this set of faults can cause any possible erroneous behavior, and so this is a "worst-case" study. However, as the scale of integrated circuit technology becomes larger, it grows increasingly difficult to postulate a restricted class of faults which would contain all faults that one might encounter in real systems. Obviously a system subject to any fault can give no information about what its correct output should be. Therefore, for the diagnosis of unrestricted faults, it is crucial that the detector D observe the input to S directly.

This immediately suggested one possible configuration for D, namely that D be an identical replica of S, running

in parallel, plus appropriate disagreement detectors. However, we investigated the question of whether one could not devise a (D,k) detector for S which is less complex than S itself. We were able to prove this is impossible. If we take number of reachable states as a measure of system complexity, we showed that if a system is on-line diagnosable for the unrestricted set of faults, then the detector must be at least as complex as the original system. Moreover, this result is true even for an arbitrarily large time delay k .

One subset of systems which was given special investigation was the class of information lossless systems. Intuitively, an information lossless system is a system with an inverse. That is, $\overline{S}$ is an inverse for S if $\overline{S}$, given the output of S as input, produces the input of S as output, with some fixed time-delay n. In this case, S is lossless. If no such $\overline{S}$ exists, S is lossy. The advantage of lossless systems is that they permit the use of loop checks, (i.e., comparing the output of the inverse system to the original input. A common example of a loop check is the use of multiplication to check division.) Further, it may be that the inverse system is less complex than the original system.

Our investigation showed that an inverse system can always be used for unrestricted-fault diagnosis if it too is information lossless. This condition is sufficient, but not necessary. Under certain conditions a lossy

inverse will also suffice. Of course, not every system has an inverse, let alone a lossless one. However, we have shown that every system has a realization to which this scheme can be successfully applied. A detailed discussion of the above results can be found in [14] and [15].

The second general class of faults that we investigated was the set of unrestricted component faults. This set arises naturally from the study of systems which can be decomposed and represented as networks of resettable state machines. Informally, an unrestricted component fault is a fault which affects only a single component machine, but which may affect that component in an unrestricted manner.

Under certain conditions such a network can be diagnosed by a combinational (e.g., no delay elements) detector, and we determined what the necessary and sufficient conditions were. We were further able to show that any network could be transformed into a combinationally diagnosable network by the addition of a single component whose complexity was no greater than the most complex component in the original network. We were also able to obtain a lower bound on the complexity of any component which will make the original network combinationally diagnosable. Detailed statements and proofs of these results can be found in [15].

2.2.3 Topics for Further Investigation

Although much progress has been made toward achieving a thorough understanding of on-line diagnosis, there are many areas that deserve further investigation. Our study to date has dealt with generally unstructured systems. While such an approach is well suited to the development of formal concepts, and to the identification of important parameters, certain questions can be better answered in a more structured environment. One reason for this is that, with a structured system, we can consider restrictions on the causes of faults. For example, given an abstract system it makes no sense to speak of faults caused by bridging failures. However, given a circuit diagram of a specific system, for example, we can discuss specific types of failures and determine the corresponding faults.

There are many different structural levels that could be explored in a meaningful way. Two levels that we feel to be particularly promising are the binary state-assigned level and the logical circuit level. A system is said to be binary state-assigned if the state set $Q = \{0,1\}^n$ for some positive integer n. In particular, the problem of memory failures in such a system has obvious relevance to the subject of digital computers. This topic has been considered in the context of fault tolerance and off-line diagnosis by Meyer [15] and Yeh [16]. Only a limited amount of structure is needed to discuss such

faults, and thus they can be analyzed before the circuit design of the machine is finalized. Also, having isolated the memory faults, the remainder of the system is combinational, and hence inherently easier to analyze. Finally, time-space tradeoffs are possible in the diagnosis of memory faults. We have not had time to investigate this question properly, but it seems clear that there is room for much study in this area.

A system possesses structure at the logical circuit level if a representation of the system can be given in terms of a logical circuit composed of primitive logical elements. These may be of the NOR gate variety, AND-OR-NOT, threshold elements, or any of the familiar similar elements. This representation level is useful for investigating failures in the primitive components.

Further work could also be performed at the architectural level of structural detail. It is at this level that one is considering the problems of implementing on-line diagnosis on a whole computer, while at the other levels emphasis would be on diagnosing a single module. In particular, we feel that emphasis should be placed on investigating the problem of on-line fault location.

Another problem that could be studied by an extension of our present (structural level) model is the problem of automatic system reconfiguration under the control of the detector. This could be achieved by allowing for feedback from the detector to the system being observed,

and gives promise of being a fruitful area for future investigation.

Finally, in connection with general investigations of the type mentioned above, an effort must be made to examine specific subsystems of a fault tolerant computing system, such as an ALU, a RAM, a ROM, etc. Here, guided by the general techniques, one should seek specific implementations of an on-line diagnosable subsystem that is tailored to a specific use and a specified class of anticipated faults. Effort here should focus not only on the design of detectors, but also on means for augmenting the structure of the original system (a CPU, for example) to facilitate detector design and possibly improve its on-line diagnostic capability. This effort is in keeping wiht Langley-augmented objectives IV and V (see Section 1.1) and should be sustained. The effort is especially important in connection with the implementation of highly reconfigurable computer architectures, where the use of presently known duplication and coding techniques may not provide the necessary diagnostic capability.

3. REFERENCES

[1] F. P. Mathur and A. Avizienis, "Reliability analysis and architecture of a hybrid-redundant digital system: Generalized triple modular redundancy with self-repair," Proc. 1970 Spring Joint Computer Conference, AFIPS Conf. Proc., Vol. 36, 1970, pp. 375-383.

[2] F. P. Mathur, "On reliability modeling and analysis of ultra-reliable fault-tolerant digital systems," IEEE Transactions on Computers, Vol. C-20, No. 11, November, 1971, pp. 1376-1382.

[3] W. G. Boricius, W. C. Carter, and P. R. Schneider, "Reliability modeling techniques for self-repairing computer systems," Proceedings ACM 1969 Annual Conference, 1969, pp. 295-309.

[4] W. G. Boricious, W. C. Carter, D. C. Jessep, P. R. Schneider, and A. B. Wadia, "Reliability modeling for fault tolerant computers," IEEE Transactions on Computers, Vol. C-20, No. 11, November, 1971, pp. 1306-1311.

[5] R. S. Ratner, et al., "Design of a fault tolerant airborne digital computer (Vol. II - Computational requirements and technology)," Final Report, SRI Project 1406, Stanford Research Institute, Menlo Park, California, October, 1973.

[6] J. F. Meyer, "Semiannual status report, theory of reliable systems," NASA Grant NGR 23-005-622, July, 1974.

[7] J. F. Meyer, "Computation-based reliability analysis," Systems Engineering Laboratory Technical Report No. 83, The University of Michigan, Ann Arbor, January, 1975.

[8] J. F. Meyer, "Computation-based reliability analysis," Digest of the 1975 International Symposium on Fault Tolerant Computing, IEEE Computer Society, June 1975. (To appear).

[9] H. Y. Chang and J. M. Scanlon, "Design principles for processor maintainability in real-time systems," 1969 Fall Joint Computer Conference, AFIPS Proc., Vol. 35, Montvale, N.J., AFIPS Press, 1969, pp. 319-328.

[10] A. Avizienis, G. C. Gilley, F. P. Mathur, A. A. Rennels, J. A. Rohr, and D. K. Rubin, "The STAR (self-testing and repairing) computer: an investigation of the theory and practice of fault-tolerant computer design," IEEE Trans. Computers, Vol. C-20, November, 1971, pp. 1312-1321.

[11] G. R. Compton, R. S. Pringle, and P. H. Mueggler, "Effects of test capability on system reliability and availability," IEEE Trans. Reliability, Vol. R-21, November, 1972, pp. 239-244.

[12] W. W. Peterson and M. O. Rabin, "On codes for checking logical operations," IBM Journal, Vol. 3, April, 1959, pp. 163-168.

[13] F. F. Sellers, M. Hsaio and L. W. Bearnson, Error Detection Logic for Digital Computers, McGraw-Hill Book Co., Inc., New York, 1968.

[14] R. J. Sundstrom, "On-line diagnosis of sequential systems - II," Systems Engineering Laboratory Report No. 81, The University of Michigan, Ann Arbor, July, 1974. Also, "On-line diagnosis of sequential systems - III," SEL Report No. 84, January, 1975.

[15] J. F. Meyer, "Fault Tolerant Sequential Machines," IEEE Trans. on Computers, Vol. C-20, No. 10, October, 1971, pp. 1167-1177.

[16] K. Yeh, "A theoretic study of fault detection problems in sequential systems," Systems Engineering Laboratory Technical Report No. 64, The University of Michigan, Ann Arbor, June, 1972.